Computer Science - Research and Development (2019) 34:61–70
https://doi.org/10.1007/s00450-018-0394-7

SPECIAL ISSUE PAPER

Analyzing the power consumption behavior of a large scale data center

Kashif Nizam Khan[1,2] · Sanja Scepanovic[2] · Tapio Niemi[1] · Jukka K. Nurminen[1,2] · Sebastian Von Alfthan[3] · Olli-Pekka Lehto[3]

Published online: 29 May 2018

Abstract
The aim of this paper is to illustrate the use of application and system level logs to better understand scientific data center behavior and energy-spending. Analyzing a data center log of 900 nodes (Sandy Bridge and Haswell), we study node power consumption and describe approaches to estimate and forecast it. Our results include methods to cluster nodes based on different vmstat and RAPL measurements as well as Gaussian and GAM models for estimating the plug power consumption. We also analyze failed jobs and find that non-successfully terminated jobs consume around 40% of computing time. While the actual numbers are likely to vary in different data centers at different times, the purpose of the paper is to share ideas of what can be found by statistical and machine learning analysis of large amount of log data.

Keywords RAPL · Energy modeling · Energy efficiency · Data center log analysis

1 Introduction

According to a recent report by Lawrence Berkeley National Laboratory [16] the data centers in United States consumed 70 billion kWh of electricity in 2014. The consumption is predicted to grow even higher although the growth has been more moderate than expected earlier. One reason for the moderate growth of power consumption while the computing needs have drastically increased, has been the attention of both high performance computing (HPC) industry and researchers to improve the energy efficiency. Reduced consumption results both in smaller electricity bill and reduced environmental load.

Möbius et al. [13] provide a comprehensive survey of electricity consumption estimation in HPC systems. The techniques can be broadly categorized as direct measurements and power modeling. Direct measurement techniques involve power measuring devices or sensors to monitor the current draw [14] whereas power modeling techniques estimate the power draw with system utilization metrics such as hardware counters or Operating System (OS) counters [5].

Intel's Running Average Power Limit (RAPL) is one such power measurement tool, which has been useful in power measurement and modeling research [8,11,17]. RAPL reports the real time power consumption of the CPU package, cores, DRAM and attached GPUs using Model Specific Registers (MSRs). Since its introduction in Sandy Bridge it has evolved and in newer architectures, Haswell and Skylake, RAPL works as a reliable and handy power measurement tool [8].

In this paper, we study and analyze the energy consumption of a computing cluster named *Taito*, which is a part of the CSC - IT Center for Science in Finland. In *Taito*, most of the jobs come from universities and research institutes. They are typically simulation or data analysis jobs and run parallel on multiple cores and nodes. We utilize a dataset of 900 nodes (Sandy Bridge and Haswell) which includes OS counter logs from vmstat tool (see Table 1), CPU package power consumption values from RAPL and plug power consumption value sampled at a frequency of approximately 0.5 Hz over a period of 42 h (more details in Sect. 3).

The aim of this study is to show examples of information that can be extracted from data center logs. In particular we

1. Investigate how OS counters and RAPL measurements can be used to explain and estimate the total power consumption of a computing node (Sects. 4, 5 and 7).
2. Analyse failed jobs and their influence in energy spending (Sect. 6).

✉ Kashif Nizam Khan
kashifnizam.khan@gmail.com

[1] Helsinki Institute of Physics, Helsinki, Finland

[2] Aalto University, Espoo, Finland

[3] CSC - IT Center for Science, Espoo, Finland

Table 1 Vmstat output variables used: description and min and max values in CSC dataset

Vmstat variable	Description	Min.	Max.
r	# of processes waiting CPU time	0	200
b	# of processes waiting on I/O	0	97
swpd	# of virtual memory blocks	0	9,775,548
free	# of blocks of idle memory	393,316	876,866,240
cache	# of memory blocks used as cache	15,656	622,179,392
si	# of blocks per sec swapped in	0	27
so	# of blocks per sec swapped out	0	27
bi	# of blocks received from HD	0	1247
bo	# of blocks sent to from HD	0	3461
in	# of interrupts per sec	0	74
cs	# of context switches per sec	0	73
us	User time % of CPU time	1	97
sy	System (kernel) % of CPU time	0	20
id	Idle % of CPU time	2	92
wa	% of CPU time waiting for IO	0	29

3. Cluster the nodes based on the OS counter and RAPL values. This gives an indication of the opportunities to combine different workload in a way which uses the resources in a balanced way (Sect. 5).
4. Use machine learning to map power consumption to OS counter values (Sects. 7 and 8).

2 Related works

Power measurement is one of the key inputs in any energy efficient system design. As such, it has been quite extensively studied in the energy efficiency literature for HPC systems and data centers. As described in Sect. 1, the measurement techniques can be categorized as direct measurements and power modeling. Direct measurements using external power meters provide accurate measurements and can give real time power consumption of different components of the system depending on the type of hardware and software instrumentation [6,7]. However, direct measurement techniques often require physical system access and custom and complex instrumentations. Sometimes such techniques may hinder the normal operation of the data center [5].

Modern day data centers also make use of sensors and/or Power Distribution Units (PDUs) that monitor and report useful runtime information about the system such as power or temperature. Such tools also show good accuracy. However, PDUs and sensors can be costly to deploy and may not scale well as the demand increases. These devices are not yet commonly deployed and they might have usability issues as reported in [5].

Power modeling using performance counters are quite useful with regards to cost, usability and scaling. There are mainly two types of such counters which can be used in power modeling of computing systems, namely hardware performance counters (often referred as Performance monitoring counters(PMC)) and OS provided utilization counters or metrics. PMCs have been used quite extensively in monitoring the system behavior and finding correlation with power expenditure of systems thus providing a useful input for power modeling approaches [2,9]. However, such models often suffer from problems like limited number of events that can be monitored and then PMCs are often architecture dependent and so the models may not be transferable from one architecture to the other [13]. The accuracies of such models are also often workload dependent and as such may not be reliable at times [5,13].

Intel introduced the RAPL interface [10] to limit and monitor the energy usage on its Sandy Bridge processor architectures. It is designed as a power limiting infrastructure which allows users to set a power cap and as a part of this process it also exposes the power consumption readings of different domains. RAPL is implemented as Model-Specific Registers (MSRs) which are updated roughly every millisecond. RAPL provides energy measurements for processor package (PKG), power plane 0 (PP0), power plane 1 (PP1), DRAM, and PSys which concerns entire System on Chip (SoC). PKG includes the processor die that contains all the cores, on-chip devices, and other uncore components, PP0 reports the consumption of CPU cores only, PP1 holds the consumption of on-chip graphics processing units (GPU) and DRAM plane gives the energy consumption of dual in-line memory modules (DIMMs) installed in the system. From Intel's Skylake architecture onwards RAPL also reports the consumption of entire SoC in PSys domain (it may not be available on all Skylake versions). In Sandy Bridge, RAPL domain values were modeled (not measured) and thus it had some deviations from the actual measurements [8]. With the introduction of Fully Integrated Voltage Regulators (FIVRs) in Haswell, RAPL readings have promisingly improved and it has proved its usefulness in power modeling also [11].

There has also been interesting works regarding the job power consumption and estimation for data centers [3,15]. Borghesi et al. [3] proposed machine learning technique to predict the consumption of HPC system using real production data from Eurora supercomputer. Their prediction technique show an average error of approximately 9%. In our analysis, we show a different analysis of data center power consumption since we use system utilization metrics from OS counters and RAPL. Our results confirm a few of the observations already seen in literature. However, our approach is different since we make use of tools like vmstat and RAPL from a real life production dataset. We show the power consumption predictability of such tools and we pinpoint metrics which tend to correlate more with the power readings than the other as we cluster nodes based on vmstat and RAPL values. This paper also demonstrates different modeling techniques (leveraging machine learning) to model the plug power from OS counter and RAPL values and pinpoints essential parameters that influence the accuracy of such techniques.

3 Dataset description

The CSC dataset consists of around 900 nodes which are all part of *Taito* computing cluster. Among the 900 nodes, there are approximately 460 Sandy Bridge compute nodes, 397 Haswell nodes and a smaller number of more specialized nodes with GPUs, large amounts of memory or fast local disks for I/O intensive workloads. Since there are different hardwares and hence performance differences between the two types of nodes, their power consumption exhibit different patterns (see Fig. 1).

Table 2 Hardware configurations—*Taito* compute nodes

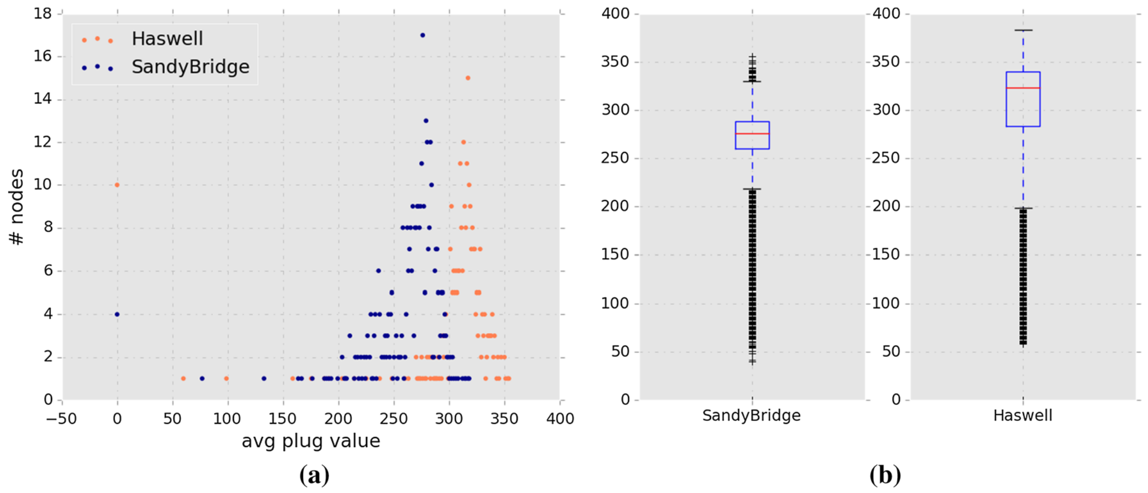
Type	Haswell	Sandy bridge
Number of nodes	397	496
Node model	HP XL230a G9	HP SL230s G8
Cores/node	24	16
Memory/node	128 GB	64 GB

The dataset, captured in June 2016, consists of vmstat output (Table 1), RAPL package power readings, plug power obtained from Intelligent Platform Management Interface (IPMI) and job ids. All of these are sampled at a frequency of approximately 0.5 Hz over a period of 42 h. The hardware configurations of *Taito*'s compute nodes are given in Table 2 [1].

vmstat (Virtual memory statistics) is a Linux tool, which reports the usage summary of memory, interrupts, processes, CPU usage and block I/O. The vmstat variables that we have used are presented in Table 1. The CSC dataset reports the energy consumption of two RAPL PKG domains for the dual socket based server systems in *Taito*. The metrics collection for this dataset was done manually. In order to continuously collect and analyze this type of data, better high-resolution energy measurement tools are needed which should ideally work in a cross-platform basis across different hardware and batch job schedulers.

4 Power consumption of computing nodes

We start by inspecting how the variable of interest: power consumption (measured directly at the plug) changes over time at different nodes. First observation is that there are



(a) **(b)**

Fig. 1 Power consumption differences on Haswell and Sandy Bridge nodes. **a** Distributions of average values per node, **b** whisker diagrams with all the values

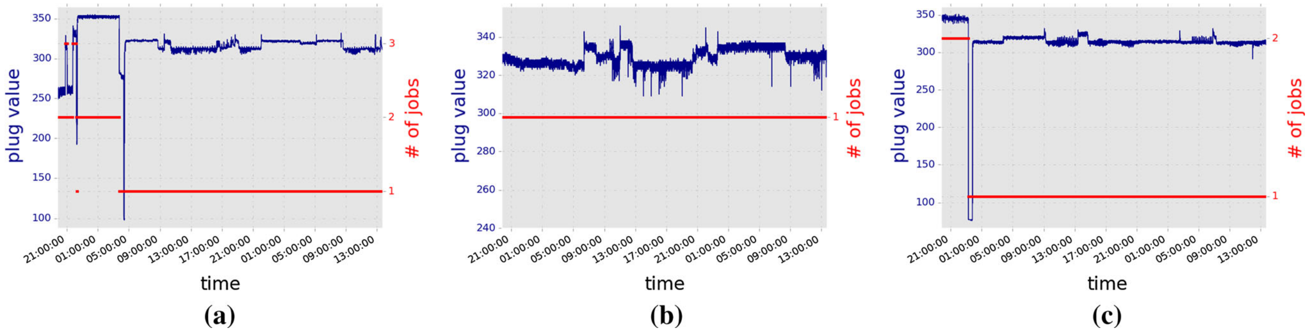


Fig. 2 Power consumption of nodes running mostly a single job. **a** Node C581, **b** node C836, **c** node C749

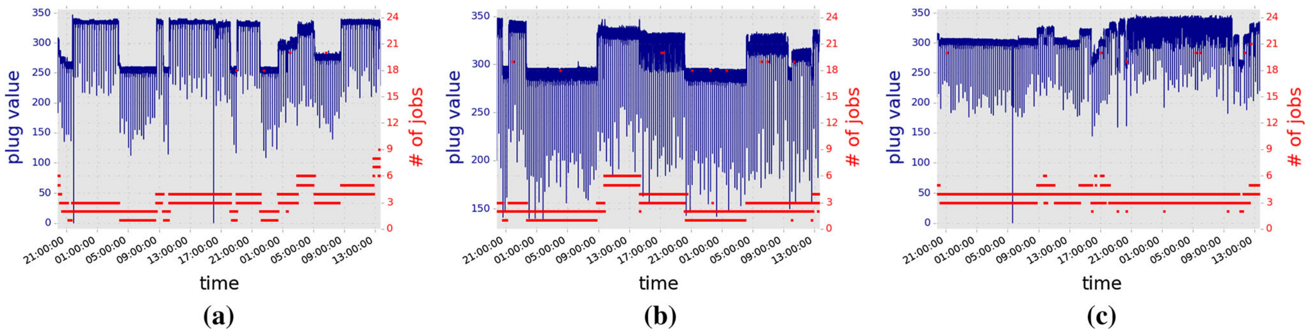


Fig. 3 Power consumption of nodes running a highly variable number of jobs. **a** Node C585, **b** node C626, **c** node C819

considerable variations in the measured power consumption between different nodes (see Fig. 1), and even at a single node, at different time intervals during the observed period. This is not surprising, as the node power consumption at any point is dependent on the type of computing jobs running on that node. In order to illustrate this variability, we show the power consumption plots of several nodes with rather diverse patterns in Figs. 2 and 3.

From Fig. 2 we observe that single running jobs also exhibit different patterns and variability in how they consume power. While the influence of the number of jobs running on a node on its power consumption is evident from Fig. 3, it is also clear that this dependency is very subtle and not straight forward to express.

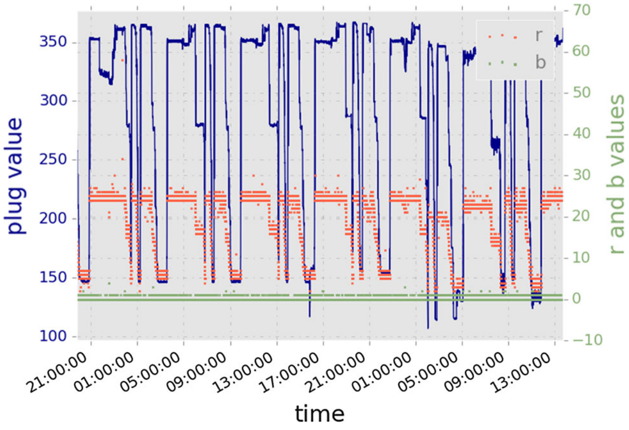
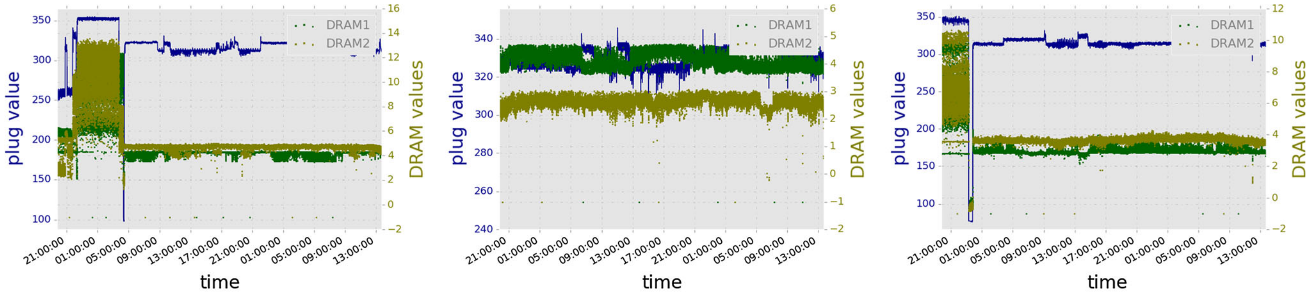


Fig. 4 Power consumption and number of user and kernel processes running—node C775 (color figure online)

5 Vmstat and RAPL variables statistics

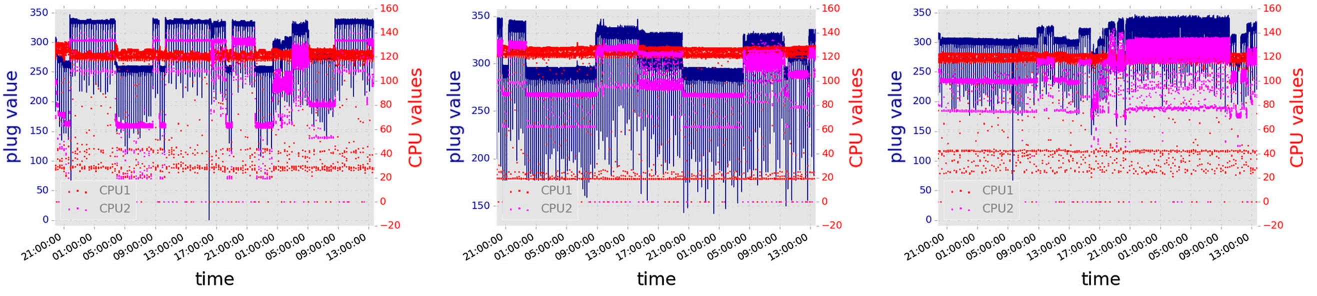
After the observations on the power consumption in relation to the number of jobs running on a node, we turn to the observation of power consumption in relation to the vmstat output values. Namely, vmstat output informs us about the consumption of different computing resources on a node and hence captures more subtle properties of the jobs running on the node. The description of the vmstat output variables in CSC dataset is presented in Table 1.

Taking the same set of nodes introduced earlier (Figs. 2 and 3), we investigate visually the interplay of vmstat and RAPL variables and power consumption. We observe that vmstat values **r,b** (see Table 1 for explanation) change even on a node running no jobs. Looking at similar analysis for the nodes running several jobs in Fig. 4, the relationship between vmstat values **r,b** and power consumption values is evident. Similarly, Fig. 5 illustrates the interplay between memory



(a) **(b)** **(c)**

Fig. 5 Power (in blue) and two types of memory consumption (see legend). **a** Node C581, **b** node C836, **c** node C749 (color figure online)



(a) **(b)** **(c)**

Fig. 6 Power (in blue) and two types of CPU consumption (see legend). **a** Node C585, **b** node C626, **c** node C819

RAPL values (DRAM) and power consumption, and Fig. 6 between CPU RAPL values and power consumption.

Figure 7 presents Self-Organizing Maps (SOM) model [12] classification output on the CSC dataset. SOM is a unsupervised classification technique to visualize high dimensional data in low dimensional space. In this figure, we cluster all the nodes in 9 clusters based on the similarity in Node data. Node count per class shows the number of nodes in different clusters as a heat map. Clusters represented in 'white' color contain around 200+ nodes whereas clusters represented in 'red' color contain around 50 or less nodes with the other colors falling in between. If we now see the same clusters in the *Node data* (left sub-figure of Fig. 7), we observe which variables dominate the similarities in that cluster. For example, *Node data* for the 'white' colored cluster in the top-right corner shows that the variables *us*, *CPU1*, *CPU2* and *plug* dominate the cluster (*CPU1*, *CPU2* correspond to the RAPL package power).

6 Analysis of unsuccessful jobs

Table 3 presents statistics of the jobs executed on the *Taito* cluster. We focus on the job exit status, number of jobs which have the same status, elapsed time per job (in hours) and total CPU Time used (user time plus system time). The dataset from *Taito* contains four types of job status: completed, failed, cancelled and timeout. Completed jobs are successful jobs that ran to completion. Failed jobs are jobs that failed to complete successfully and did not produce desirable outputs. Cancelled jobs are cancelled by their users. These are often failures but sometimes cancellation is done on purpose after the job has produced the desirable results. Timeout jobs did not run to successful completion within a given time limit. Timeouts are not necessarily failures, they are done occasionally on purpose and can produce useful outputs.

From Table 3 we can see that approximately 84% of the jobs are completed jobs and they consume 56.95% of the total CPU time. Failed jobs on the other hand constitute of 12.5% of the total jobs and they consume around 14.75% of the total CPU time. Interestingly, only 0.5% of the total jobs are timed out but they consume around 19.34% of the total CPU time. Timeout jobs also have elapsed time of 25 h per job which is by far the maximum.

If we have a pessimistic assumption that all the non-completed jobs are unsuccessful it turns out that 16% of such jobs consumed around 43% of total CPU time. This shows that the wasted resources and energy in terms of unsuccessful jobs can be as much as 43% in typical data centers. This is approximately 280.000 days of CPU time in numbers. If these failures are identified in relatively early stage of a job lifetime, the potential CPU time and energy save can be significant. It can be a potential target for energy efficiency in data center workload management.

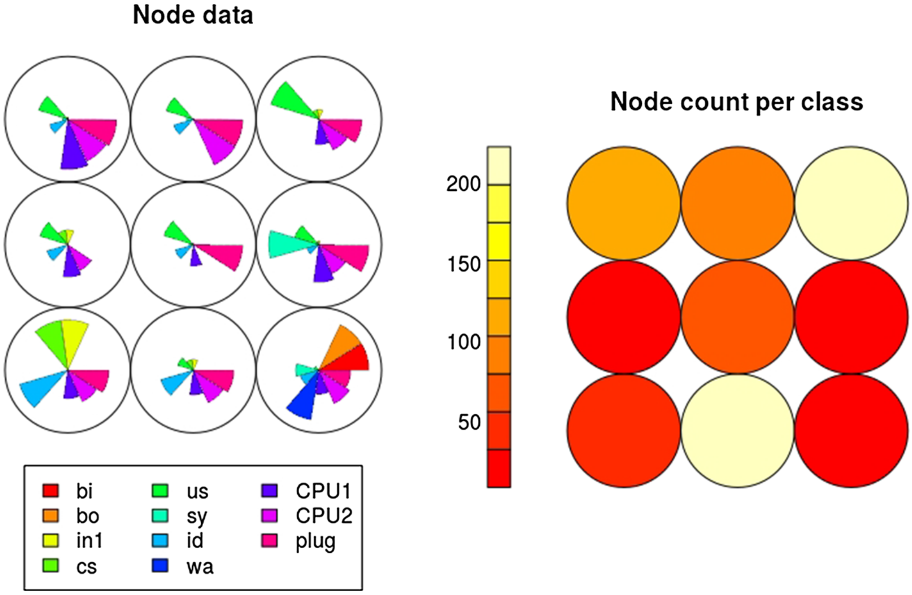


Fig. 7 Node clusters based on power, vmstat and RAPL (CPU) variables

Table 3 Job statistics—total of 809,178 jobs

Job status	Nr. of jobs (%)	Elapsed time/job (h)	CPU time (%)
Completed	84.0	1.0	56.95
Failed	12.5	0.7	14.75
Cancelled	3.0	8.0	8.96
Timeout	0.5	25	19.34

7 Estimation results

In this section we present results of power consumption estimation based on historical power consumption, vmstat and RAPL data (input to build the model) and current vmstat and RAPL values (intervention variables). We take first two-thirds of the time period (around 1 day) as historical data and we build the model on it. Afterwards we test the accuracy of prediction of such a model on the last third of the data (around half a day).

At first we tested building a model on data from a single node and predicting power at the same node. We do not report these results, as on some nodes this approach has worked rather well, but on some other nodes the results were under an acceptable limit. However, such an exercise taught us that the 'problematic' nodes on which prediction performance was poor, featured a sudden change in the patterns of power consumption and job execution during the period we were trying to predict. Since ML algorithms are designed to learn from 'seen' values, and they do not perform well on 'unseen' ones, which result in poor performance in such cases.

After such an understanding, we try building ML models on a random sample of shuffled data coming from **all** the nodes (of type Haswell) in our dataset. Precisely, we sample 2% of data from all the nodes ($251,244$ data samples) and evaluate performance of different ML algorithms on it using standard 10-fold cross validation approach. The best result is achieved using Random Forest [4] as shown in Table 4.

In addition to a high correlation coefficient, the regression model makes mean absolute error (MAE) of 3.12, which is measured in the units of target variable (power consumption). If we remind ourselves of the power consumption values on Haswell nodes in Fig. 1b, we see that such an error compared to average values around 300 yields a good result. Root mean squared error (RMSE) is more sensitive to sudden changes in the target variable, which are present in our data. Relative errors measure how well our estimation compares to a null model that would always predict the average value. The value larger than 100% would mean that our model is performing worse, while smaller values are better (Table 5).

Table 4 Power estimation: 10-fold cross validation results on a 2% sample from all nodes

Correlation coefficient (corrcoef)	0.97
Mean absolute error (MAE)	3.12
Root mean squared error (RMSE)	9.11
Relative absolute error (RAE)	12.25%
Root relative squared error (RRSE)	21.83%
Total number of instances	251,244

Table 5 Power estimation results per node

Node id	Corr. coef.	RMSE	MAE	# Instances
C832	0.92	13.92	10.84	15,838
C907	0.93	16.98	10.68	26,149
C836	0.79	1.37	1.94	27,962
C775	0.99	6.68	12.09	26,756
C581	0.96	1.60	2.00	28,136
C585	0.91	6.42	13.17	28,174
C626	0.74	6.60	13.80	28,594
C742	0.99	10.64	14.01	19,727
C749	0.68	3.34	3.73	28,197
C819	0.73	10.62	16.14	27,505

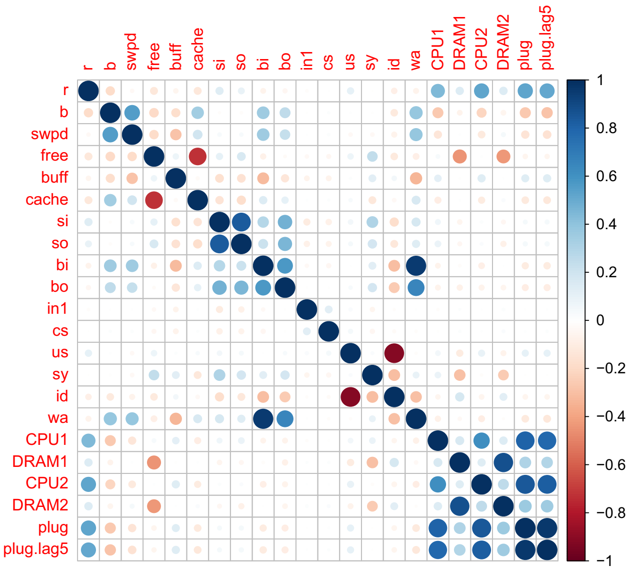


Fig. 8 Original correlation matrix

8 Modeling plug power

We take a sample of 30,000 measurements focusing on the ‘Haswell’ type computing nodes. 80% of this is used for the training set and 20% for the test set.

We aim at modelling the plug power using both OS counters and RAPL measurements. The variables and their linear correlations are shown in Fig. 8.

The distribution of the plug variable is shown in Fig. 9. The distribution does not match very well with any common theoretical distribution. However, using the normal distribution gives the best results when using regression models. We also tested whether there is any lag between the RAPL values and the plug power values and found out the best results are received when using the plug values 10 s after the RAPL measurements. The variable is named ‘lag5’, since we used 0.5 Hz sampling frequency.

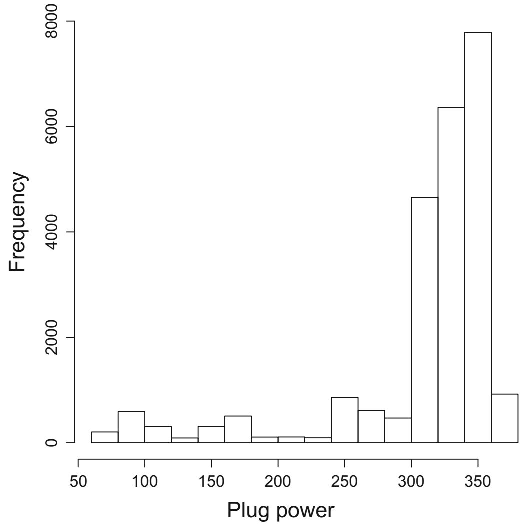


Fig. 9 Distribution of plug variable

We first fitted a linear model for estimating the plug power consumption using the RAPL parameters.

$$f(x) = a_0 + a_2CPU1 + a_3CPU2 + a_4DRAM1 + a_5DRAM2 + e \tag{1}$$

Fitting the model to our training set gave the following result:

When testing the accuracy using the test sample, the linear model gave 2.10% mean absolute percentage error. Next, we applied generalized additive models (GAM):

$$g(u) = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) + e.$$

Where x_i are covariates, β_0 the intercept, f_i smooth functions, e_i the error terms, and $g()$ the link function. This makes it possible to model non-linear relationships in a regression model. We use the same covariants as above and no link function. The mean absolute percentage error slightly decreased to 1.97%. Figure 10 shows the smooth functions of each independent variable in the GAM model. As we can see, the effect of the DRAM is much smaller than the effect of CPU. The curves are not totally linear meaning that the effect of RAPL values to the plug power is not exactly linear.

Finally, we include possible interactions among the RAPL variables into the model meaning that 2 or 3 variables can have a common effect. For example, CPU1 and DRAM2 together could increase the plug power more than both of them as separate components do. This is not included in the previous models.

Figure 11 illustrates the accuracy of the model. Large values match very well but the model has difficulties to estimate very small values. The mean absolute percentage error was slightly smaller again, 1.87%.

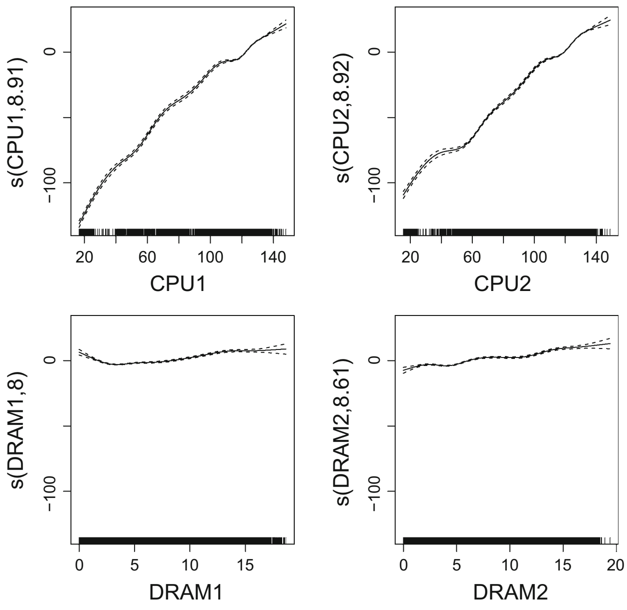


Fig. 10 Smooth functions of the GAM model

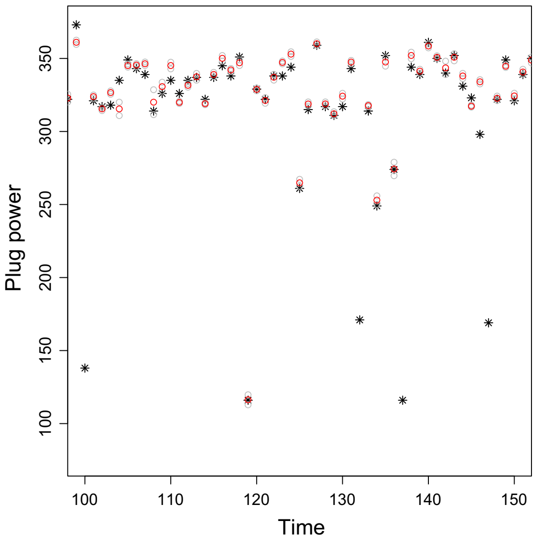


Fig. 11 Testing the GAM model with interactions against the test set. (Black asterisk = real value, red circle = estimated value, grey circles are 95% CI for the estimation) (color figure online)

In Figs. 12 and 13, we see plots illustrating the combined effects among variables. The total effect to the power consumption is shown in z-axis (upwards) while x- and y-axis represent the values of the variables. For example, in Fig. 12, we have an example of combined effect of CPU1 and CPU2 to the total power consumption. We see that the effect of CPU1 to the total power consumption decreases when its value increases, and when both the CPUs run at medium power, the total effect is slightly higher. In any case the combined effects are relative small compared to direct effects (e.g. Fig. 10).

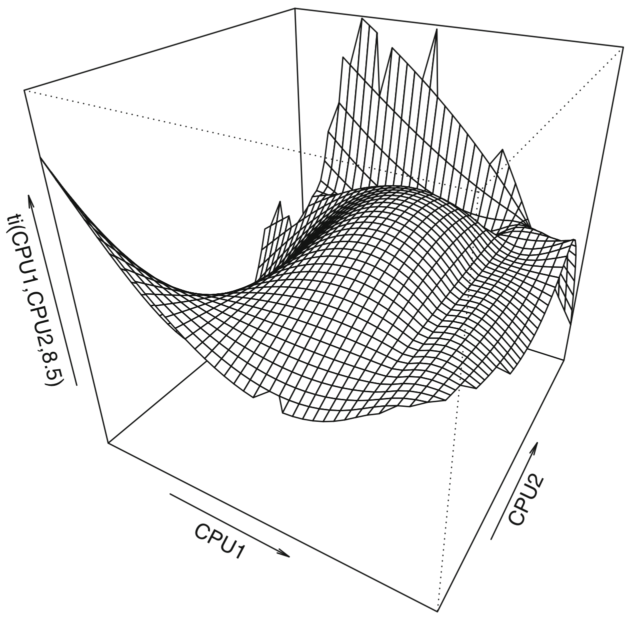


Fig. 12 The combined effect of CPU1 and CPU2 measurements. In the middle range of the both values the actual effect seems to increase

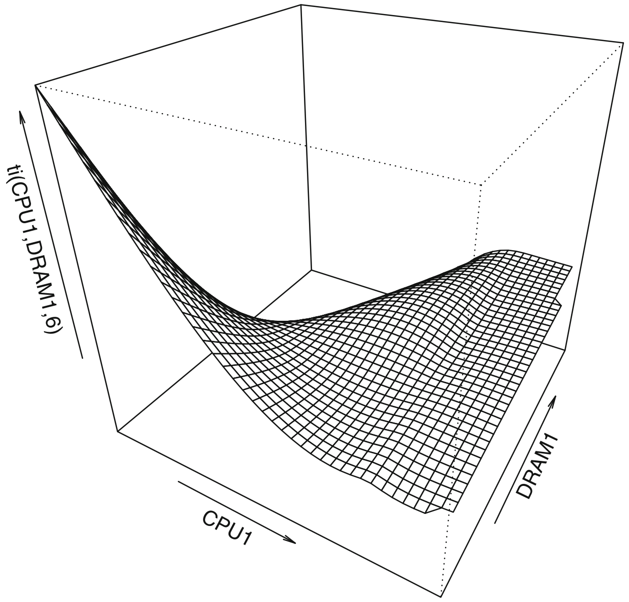


Fig. 13 The combined effect of CPU1 and DRAM1 measurements. When little DRAM power is used, the CPU power has large effect

9 Conclusion

In this paper we have presented different approaches for analyzing data center power and OS counter based utilization logs. We have shown that estimating plug power from utilization metrics is promising and the logs can be used in different ways for producing effective power models for data centers. Tools such as RAPL add to the accuracy of the models by providing real time power consumption data. For example, the GAM model shows that RAPL values can predict the plug power with mean absolute error rate of 1.97%. If we consider interactions among RAPL variables the error reduces to 1.87%. Apart from modeling, our analysis also shows that unsuccessful jobs can consume significant resources and power. If the problems can be identified early in job life cycle, resource and energy waste can be reduced. In the future, we aim to utilize such data center logs to produce job specific power consumption models and identify power consumption anomalies within data center workload management.

Acknowledgements Author Kashif Nizam Khan would like to thank Nokia Foundation for a grant which helped to carry out this work.

References

1. Taito supercluster. https://research.csc.fi/csc-s-servers/taito. Accessed 17 Marc 2017
2. Bircher WL, John LK (2012) Complete system power estimation using processor performance events. IEEE Trans Comput 61(4):563–577. https://doi.org/10.1109/TC.2011.47
3. Borghesi A, Bartolini A, Lombardi M, Milano M, Benini L (2016) Predictive modeling for job power consumption in HPC systems. Springer, Cham, pp 181–199
4. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
5. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: a survey. IEEE Commun Surv Tutor 18(1):732–794
6. Economou D, Rivoire S, Kozyrakis C, Ranganathan P (2006) Full-system power analysis and modeling for server environments. In: International symposium on computer architecture-IEEE
7. Ge R, Feng X, Song S, Chang HC, Li D, Cameron KW (2010) Powerpack: energy profiling and analysis of high-performance systems and applications. IEEE Trans Parallel Distrib Syst 21(5):658–671
8. Hackenberg D, Schöne R, Ilsche T, Molka D, Schuchart J, Geyer R (2015) An energy efficiency feature survey of the Intel Haswell processor. In: 2015 IEEE international parallel and distributed processing symposium workshop, pp. 896–904. https://doi.org/10.1109/IPDPSW.2015.70
9. Hirki M, Ou Z, Khan KN, Nurminen JK, Niemi T (2016) Empirical study of the power consumption of the x86-64 instruction decoder. In: USENIX workshop on cool topics on sustainable data centers (CoolDC 16). USENIX Association, Santa Clara, CA
10. Intel: Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A, 3B & 3C): System Programming Guide (2014)
11. Khan KN, Ou Z, Hirki M, Nurminen JK, Niemi T (2016) How much power does your server consume? Estimating wall socket power using RAPL measurements. Comput Sci Res Dev 31(4):207–214
12. Kohonen T (1998) The self-organizing map. Neurocomputing 21(1):1–6
13. Möbius C, Dargie W, Schill A (2014) Power consumption estimation models for processors, virtual machines, and servers. IEEE Trans Parallel Distrib Syst 25(6):1600–1614
14. Molka D, Hackenberg D, Schöne R, Müller MS (2010) Characterizing the energy consumption of data transfers and arithmetic operations on x86-64 processors. In: International conference on green computing, pp 123–133
15. Podzimek A, Bulej L, Chen LY, Binder W, Tuma P (2015) Analyzing the impact of cpu pinning and partial cpu loads on performance and energy efficiency. In: 2015 15th IEEE/ACM international symposium on cluster, cloud and grid computing, pp 1–10. https://doi.org/10.1109/CCGrid.2015.164
16. Shehabi A, Smith S, Horner N, Azevedo I, Brown R, Koomey J, Masanet E, Sartor D, Herrlin M, Lintner W (2016) United states data center energy usage report. Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775, p 4
17. Zhai Y, Zhang X, Eranian S, Tang L, Mars J (2014) HaPPy: hyperthread-aware power profiling dynamically. In: 2014 USENIX annual technical conference (USENIX ATC 14), pp 211–217. USENIX Association, Philadelphia, PA

Kashif Nizam Khan has completed his PhD in Computer Science from Aalto University in May, 2018. During his PhD he was working in the Green-BigData project with Helsinki Institute of Physics and Department of Computer Science, Aalto University. He obtained his Masters in Security and Mobile Computing from Aalto University, Finland and NTNU, Norway (2008–2010). His current research focus is on energy efficiency in data-centers, big-data processing frameworks and network virtualization.

Sanja Scepanovic is completing her Ph.D. in data science at Aalto University in Finland. She is part of the Data Mining group led by Prof. Aristides Gionis. Her doctoral thesis co-supervisor is Prof. Pan Hui. She analyzes large-scale datasets towards increasing the scientific knowledge in the fields of social networks, human mobility, Web cyber security and smart energy grid. In parallel, she is enrolled in the business and entrepreneurship degree at the EIT Digital Academy. She visited CERN, TU Delft and is also an alumna of International Space University Her professional experiences include governmental institutions in Montenegro and a startup in Silicon Valley. Her interest is data science in general, with a particular focus on computational social science and the application of data science for the space domain.

Tapio Niemi is senior scientist at HEC, University of Lausanne, Switzerland. He holds a Ph.D. in computer science from the University of Tampere (2001). Between 2004 and 2017 he worked at CERN in grid, cloud and green computing projects. His research interests are data science, green computing, data management, and business intelligence. He also has good knowledge of distributed and parallel big data processing.

Jukka K. Nurminen is a principal scientist at VTT and an adjunct professor of computer science at Aalto University. In 2011–2015 he spent 5 years as a professor at Aalto working on mobile cloud computing and energy-efficiency. He has a strong industry background with almost 25 years experience of software research at Nokia Research Center. Jukkas experience ranges from mathematical modeling to expert systems, from network planning tools to solutions for mobile phones, and from R&D project management to tens of patented inventions. Jukka received his M.Sc degree in 1986 and Ph.D. degree in 2003 from Helsinki University of Technology. His research interest are focused on systems and solution that are energy-efficient, distributed, or mobile.

Sebastian von Alfthan is a Senior Application Specialist at CSC - IT Center for Science Ltd., and is there heading the procurement of a new innovative HPC infrastructure, as well as doing specialist tasks in HPC code development. Before that he also worked as the main developer of a space plasma simulation code at the Finnish Meteorological Institute. Sebastian received his Masters degree from the Helsinki University of Technology in 2002, and his Ph.D. in 2006.

Olli-Pekka Lehto is currently a Systems Administrator at Jump Trading Pacific Pte. Ltd. He has previously worked at CSC - IT Center for Science Ltd. as the manager of the Computing Platforms group and in a variety of specialist roles related to high-performance computing. Olli-Pekka received his Masters degree in networking technology from the Helsinki University of Technology in 2007.